



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

Wakamori Chiaki  
3/16/2022



# 概要

---

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

# Executive Summary

---

- 方法論のまとめ

  - データの収集

  - データラングリング

  - SQL、Pandas、Matplotlibを用いた探索的データ分析

- インタラクティブなビジュアル分析およびダッシュボード

  - 予測分析の概要

  - 結果の概要

  - 探索的データ分析結果

  - インタラクティブな分析デモ

  - 予測分析結果

# Introduction

---

- プロジェクトの背景と経緯

**SpaceX**社が多数の打ち上げを可能にする理由の**1**つは、ロケットの打ち上げ費用が比較的安価である

**SpaceX**のウェブサイトでは、**Falcon 9**ロケットの打ち上げ費用は**6200**万ドルとになっているが、他のプロバイダーではそれぞれ**1億6500**万ドル以上かかっている。これは**SpaceX**社が第**1**段を再利用できるため、打ち上げ費用が低くなっている。従って、第**1**段が着陸するかどうかを判断できれば、打ち上げのコストを判断することができる。

- 解決したい問題

**SpaceX**社がロケットの第**1**段の着陸に成功するかどうかを、データを使って予測する



Section 1

# Methodology

# Methodology

---

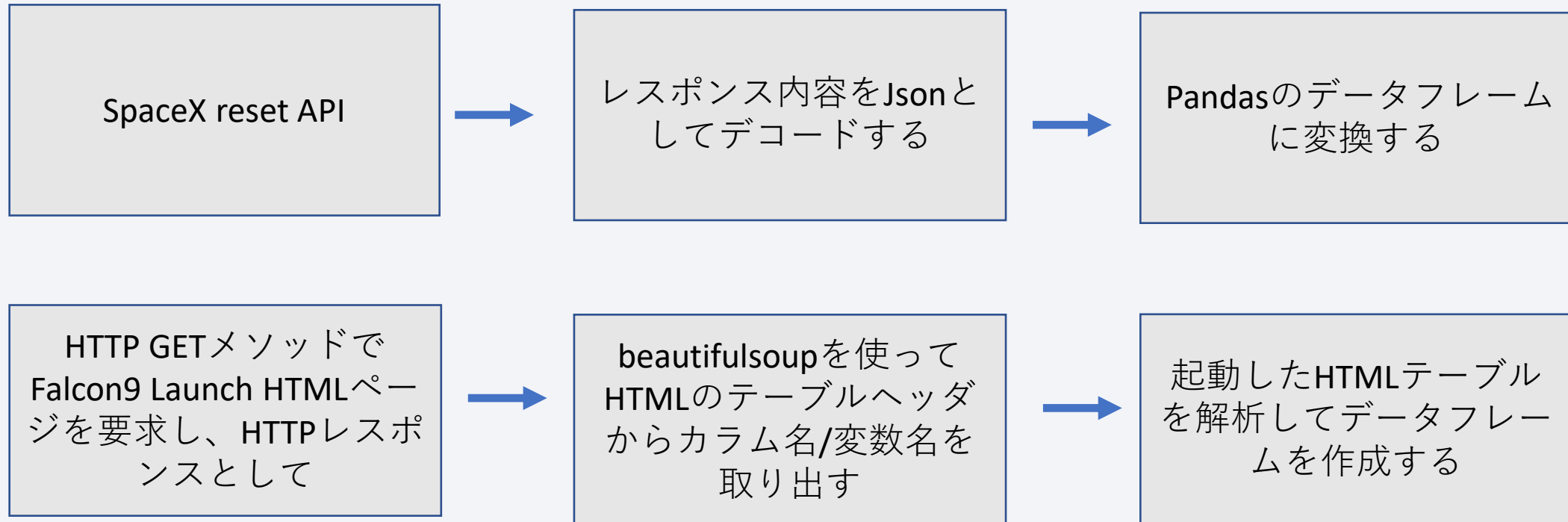
## Executive Summary

- データ収集の方法：
  - SpaceX REST API
  - Wikipediaからのウェブスクレイピング
- データラングリングの実行
  - 探索的データ解析（**EDA**）により、データからいくつかのパターンを見つけ、教師ありモデルをトレーニングするためのラベルを決定する。
- 可視化と**SQL**を使用した探索的データ分析（**EDA**）の実行
- **Folium**と**Plotly Dash**を使用したインタラクティブなビジュアル分析の実行
- 分類モデルを使った予測分析の実行
  - **SVM**, 分類木, ロジスティック回帰に最適なハイパーパラメータを探す

# Data Collection

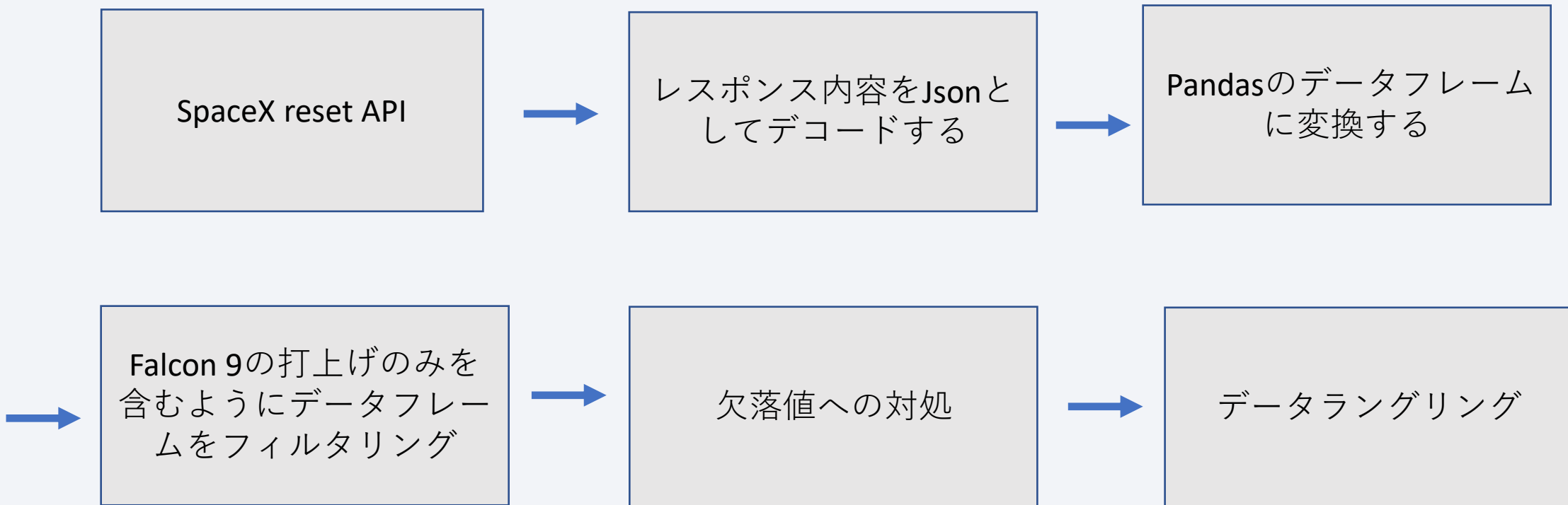
---

- データセットの収集方法 SpaceX Reset APIとWikipediaのWeb Scrapingからデータを収集する。



# Data Collection – SpaceX API

データセットの収集方法 SpaceX Reset APIとWikipediaのWeb Scrapingからデータを収集する。

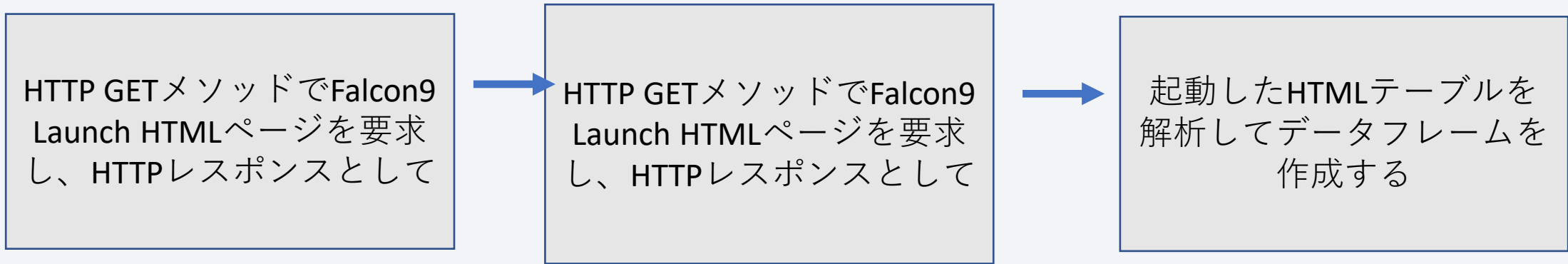




# Data Collection - Scraping

---

WikipediaからFalcon 9とFalcon Heavyの発射記録をWebスクレイピング

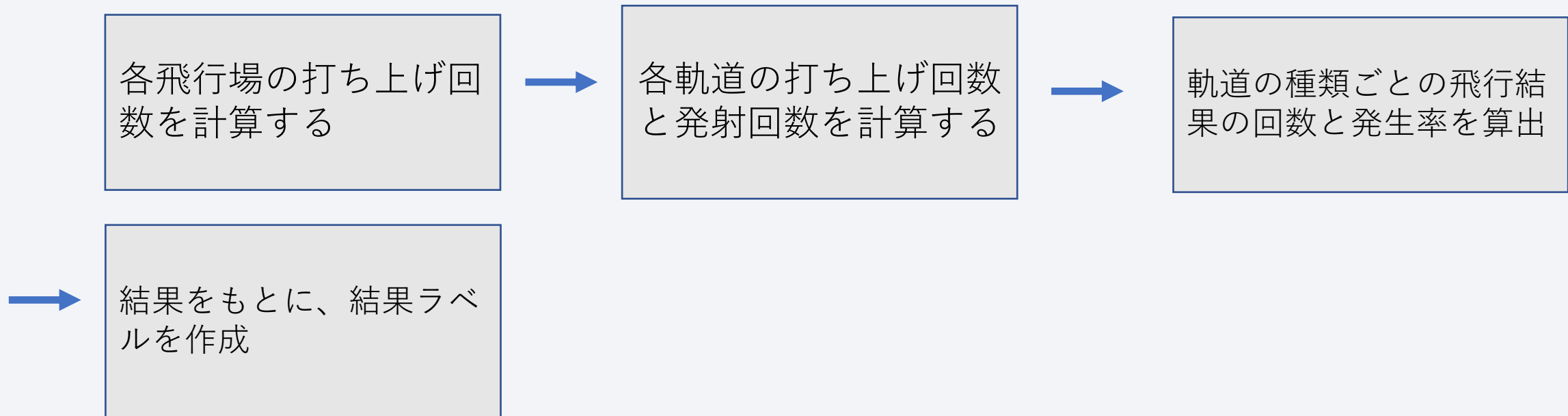


[https://github.com/waka1234/IBM\\_WatsonStudio/blob/master/jupyter-labs-webscraping.ipynb](https://github.com/waka1234/IBM_WatsonStudio/blob/master/jupyter-labs-webscraping.ipynb)

# Data Wrangling

---

探索的データ解析（EDA）を行い、データのパターンを見つけ、教師ありモデルを訓練するためのラベルを決定する。



# EDA with Data Visualization

---

関数 `catplot` を使って以下の関係を図で表した

フライトナンバーとローンチサイトの関係

打ち上げ場所とペイロード質量との関係

成功率と軌道の種類との関係成功率と軌道の種類との関係

フライトナンバーと軌道タイプのプロット関係

ペイロードと軌道の種類との関係

# EDA with SQL

---

- 宇宙ミッションに登場する固有の発射場の名前を表示する

```
> %sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL
```

- 発射場が文字列'**CCA**'で始まるレコードを**5**件表示する

```
> %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

- **NASA(CRS)**が打ち上げたブースターの総搭載質量が表示させる

```
> %sql SELECT SUM(PAYLOAD_MASS_KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'
```

- ブースターバージョン**F9 v1.1**が搭載する平均ペイロード質量を表示

```
> %sql SELECT AVG(PAYLOAD_MASS_KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1'
```

- 地上パッドへの着陸に初めて成功した日を表示

```
> %sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME='Success (ground pad)'
```

- ドローンシップに成功したブースターのうち、ペイロード質量が**4000**以上**6000**未満であるブースターの名称を列挙する

```
> %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS_KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone ship)'
```

- 成功したミッションと失敗したミッションの合計数をリストアップ

```
> %sql SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
```

- 最大積載量を運んだブースター・バージョン名をリストアップ

```
> %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS_KG_ = (SELECT MAX(PAYLOAD_MASS_KG_) FROM SPACEXTBL)
```

- サブクエリを使用する2015年の月の月名、ドローン船、ブースターバージョン、発射場の着陸失敗結果を表示

```
> %sql SELECT TO_CHAR(TO_DATE(MONTH("DATE"), 'MM'), 'MONTH') AS MONTH_NAME,
    ¥ LANDING__OUTCOME AS LANDING__OUTCOME, ¥ BOOSTER_VERSION AS BOOSTER_VERSION,
    ¥ LAUNCH_SITE AS LAUNCH_SITE ¥

FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015%'
```

- 日付2010年04月06日から2017年03月20日の間に成功した着陸\_成果のカウントを降順でランク付け

```
> %sql SELECT "DATE", COUNT(LANDING__OUTCOME) as COUNT FROM SPACEXTBL ¥
WHERE "DATE" BETWEEN '2010-06-04' and '2017-03-20' AND LANDING__OUTCOME LIKE '%Success%' ¥
GROUP BY "DATE" ¥ ORDER BY COUNT(LANDING__OUTCOME) DESC
```



# Build an Interactive Map with Folium

---

- 地図上にすべての打ち上げ場所をマークする
  - `folium`を使用して、特定の座標にテキストラベル付きのハイライトされた円形領域を追加
  - サイトマップの各ロケーションに`folium.Circle`と`folium.Marker`を作成し、追加
- 地図上の各サイトの打ち上げの成功/失敗をマークする`launch_sites`データフレームに
  - `marker_color`という新しいカラムを作成し、クラス値に基づいたマーカーの色を保存
  - `spacex_df`データフレームの各打ち上げ結果に対して、`marker_cluster`に`folium.Marker`を追加
- 発射場からその近傍までの距離を算出
  - `MousePosition`を追加して、地図上でマウスオーバーしたときの座標（`Lat, Long`）を取得する
  - `MousePosition`を使って最も近い海岸線の点をマークダウンし、その海岸線の点と発射地点との間の距離を計算

# Build a Dashboard with Plotly Dash

---

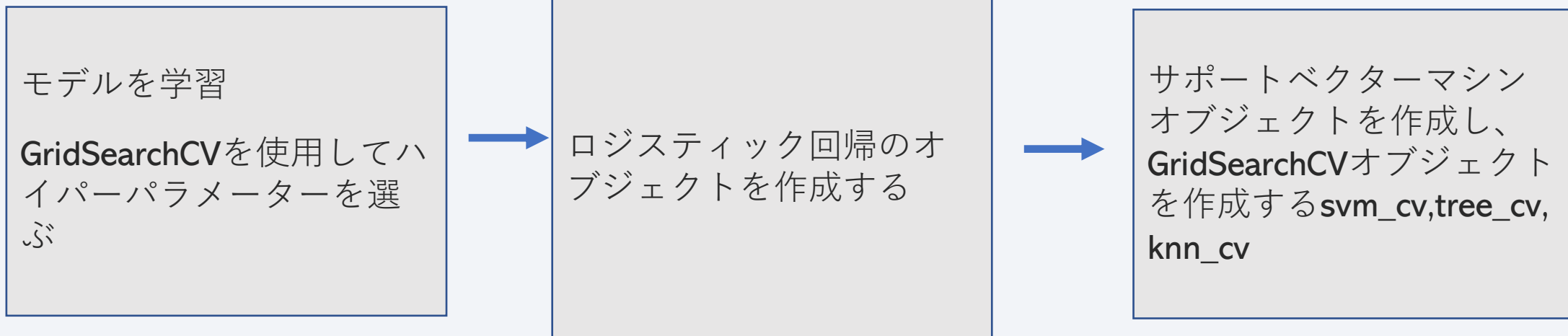
SpaceX社の打上げデータをリアルタイムでインタラクティブにビジュアル分析するための**Plotly Dash**アプリケーションの構築

- ・ 打上げ場所のドロップダウン入力コンポーネントを追加
- ・ 選択されたサイトドロップダウンに基づき、サクセスピチャートを表示するコールバック関数を追加
- ・ ペイロードを選択するためのレンジスライダーを追加
- ・ **success-payload-chart**の散布図を描画するコールバック関数を追加

# Predictive Analysis (Classification)

---

機械学習パイプラインを作成  
データがあれば打ち上げ予測



[https://github.com/waka1234/IBM\\_WatsonStudio/blob/master/IBM-DS0321EN-SkillsNetwork\\_labs\\_module\\_4\\_SpaceX\\_Machine\\_Learning\\_Prediction\\_Part\\_5.jupyterlite.ipynb](https://github.com/waka1234/IBM_WatsonStudio/blob/master/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb)



The background of the slide is an abstract composition. It features a solid blue area on the left side, which transitions into a complex pattern of diagonal streaks and lines in shades of blue, red, and cyan on the right. These streaks have a textured, almost woven appearance, suggesting a digital or data-driven theme. The overall effect is dynamic and modern.

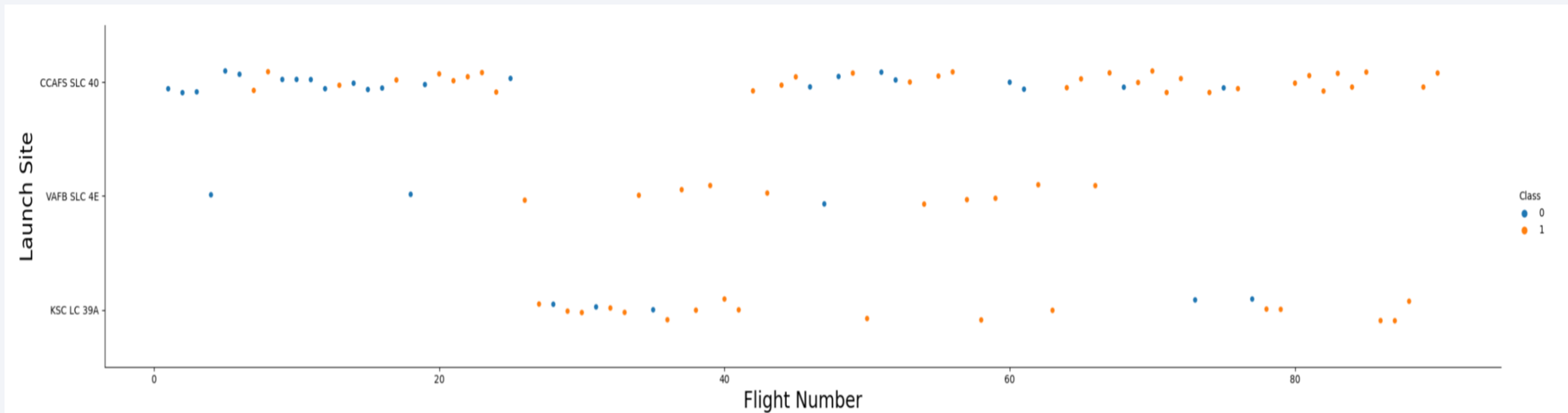
Section 2

# Insights drawn from EDA



# Flight Number vs. Launch Site

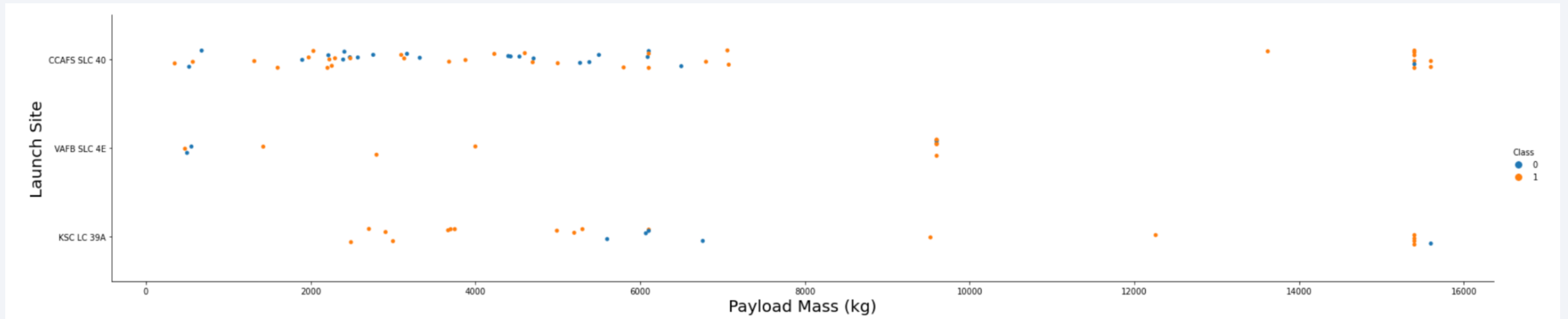
- VAFB SLC 4EとKSC LC 39Aの成功率が高い
- CCAFS SLC 40は、フライト数が上がると成功率も上昇する





# Payload vs. Launch Site

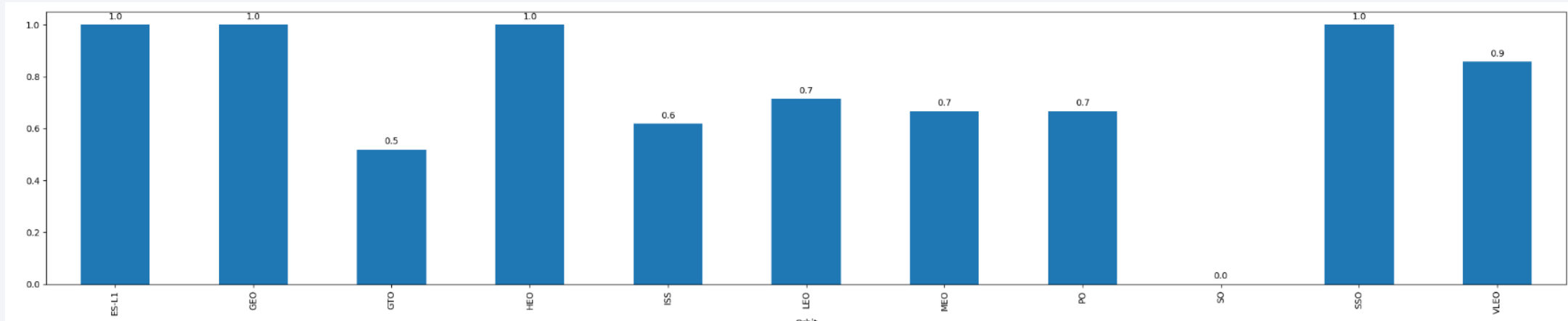
- ペイロード質量が**8000kg**を超えると成功率が高くなります。
- KSC LC 39Aは、ペイロード質量が**6500～7500kg**のときに失敗率が高くなる。
- VAFB SLC 4Eは成功率が高い
- CCAFS SLC 40は、ペイロード質量が**8000kg**未満では成功/失敗率はあまり変わらないが、**12000kg**以上では成功率が高くなる



# Success Rate vs. Orbit Type

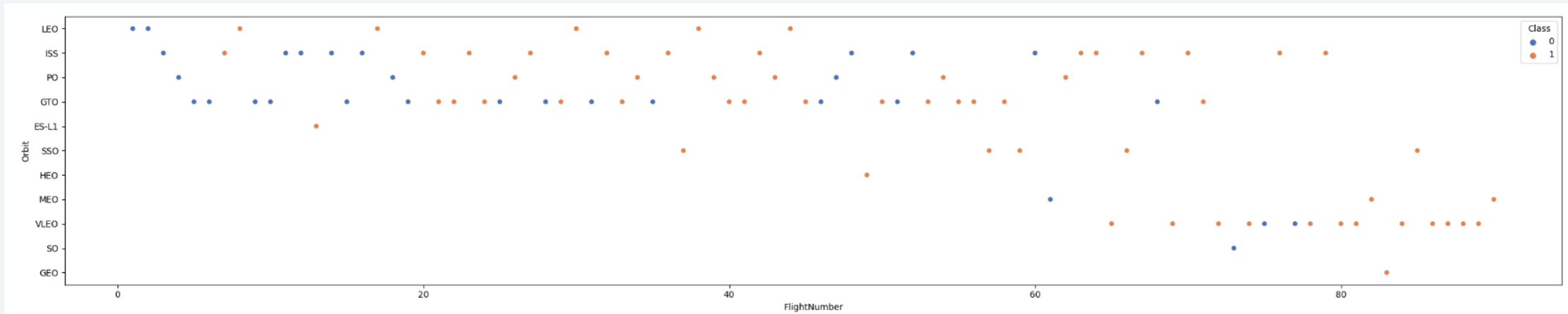
---

- ES-L1、GEO、HEO、SSOは100%の成功率
- GTO、ISS、SO以外は成功率が高い
- SOは成功率0%です



# Flight Number vs. Orbit Type

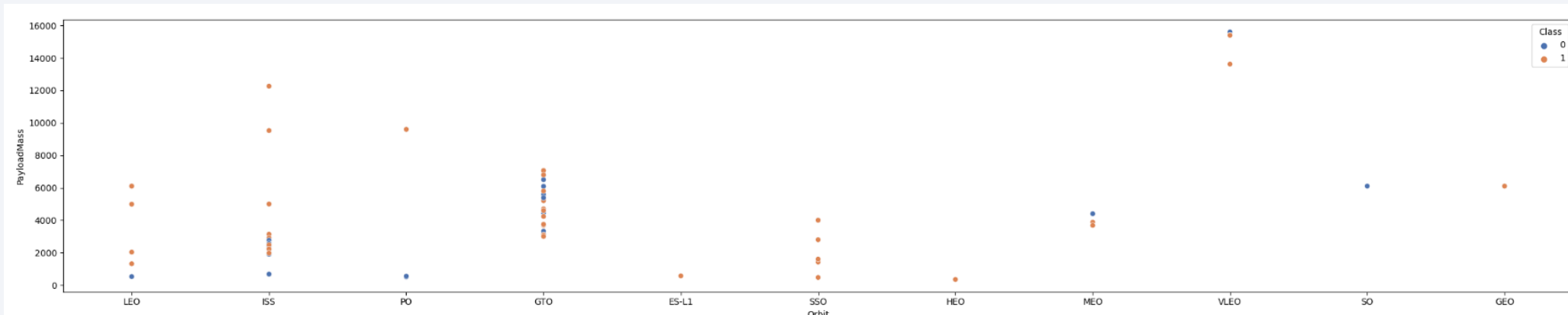
- 成功率100%はFlight NumberのOrbitデータが1つしかない
- Flight Numberが大きくなるにつれて、成功率が高くなる傾向がある



# Payload vs. Orbit Type

---

- LEO、ISSの場合、ペイロード質量が大きいほど成功率は高くなる
- GTOの場合、ペイロードの質量が大きいほど成功率は低くなる



A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

Section 3

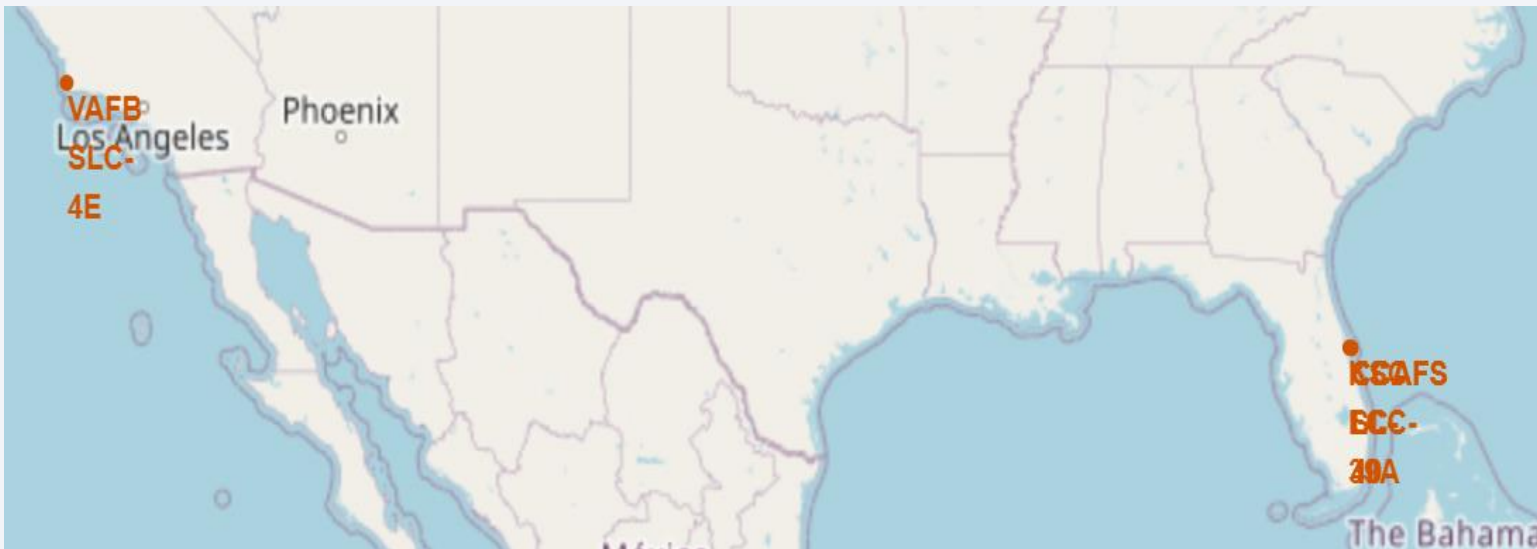
# Launch Sites Proximities Analysis



# <Folium Map Screenshot 1>

---

下図は**SpaceX**社の打ち上げデータから作成した地図で、打ち上げ場所をマークしている



## <Folium Map Screenshot 2>

以下の画像は、発射地点の位置を示している  
赤いマークは飛行機打ち上げ成功、緑のマークは飛行機打ち上げ失敗を示す

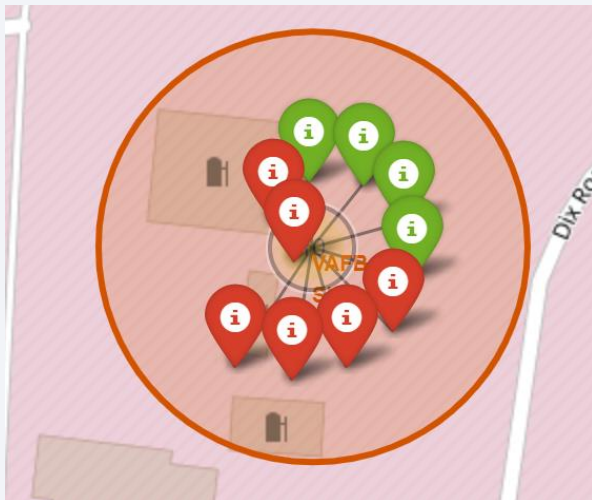


Fig.1 Launch site in VAFB SLC-4E

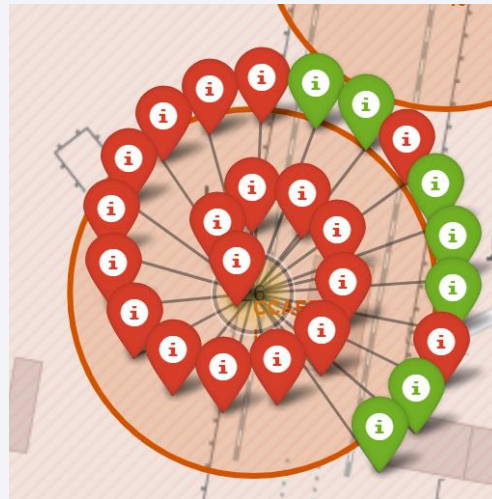


Fig.2 Launch site in CCAFS LC-40

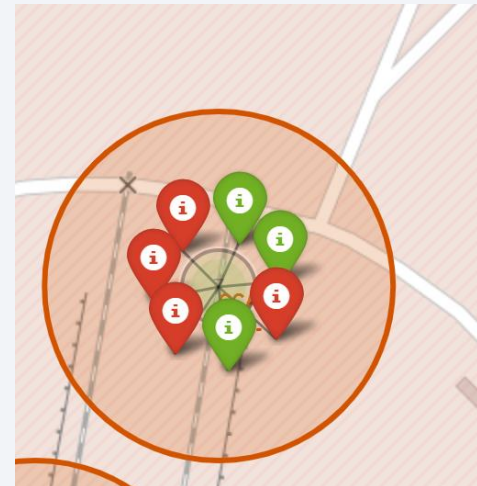


Fig.3 Launch site in CCAFS SLC-40

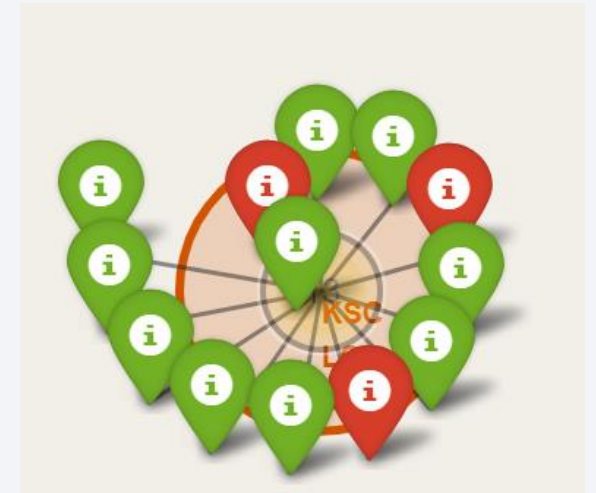
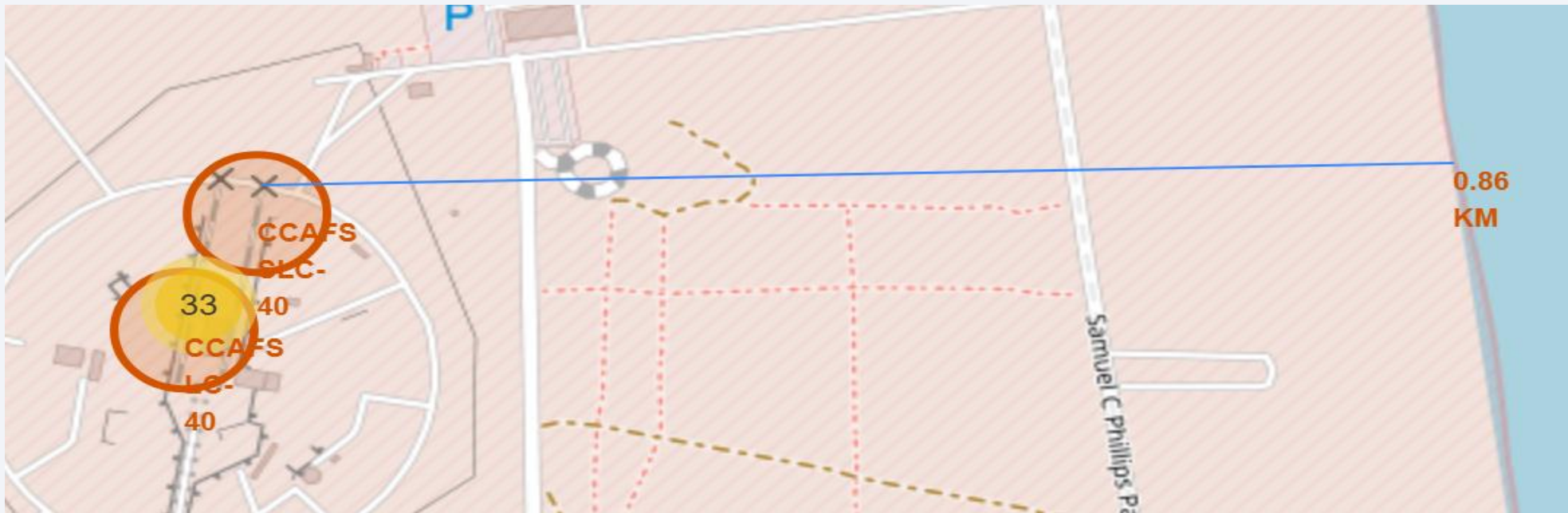


Fig.4 Launch site in KSC LC-9A

## <Folium Map Screenshot 3>

下図は、最も近い海岸線をマークし、その海岸線ポイントから発射場までの距離を計算した結果を示したものの





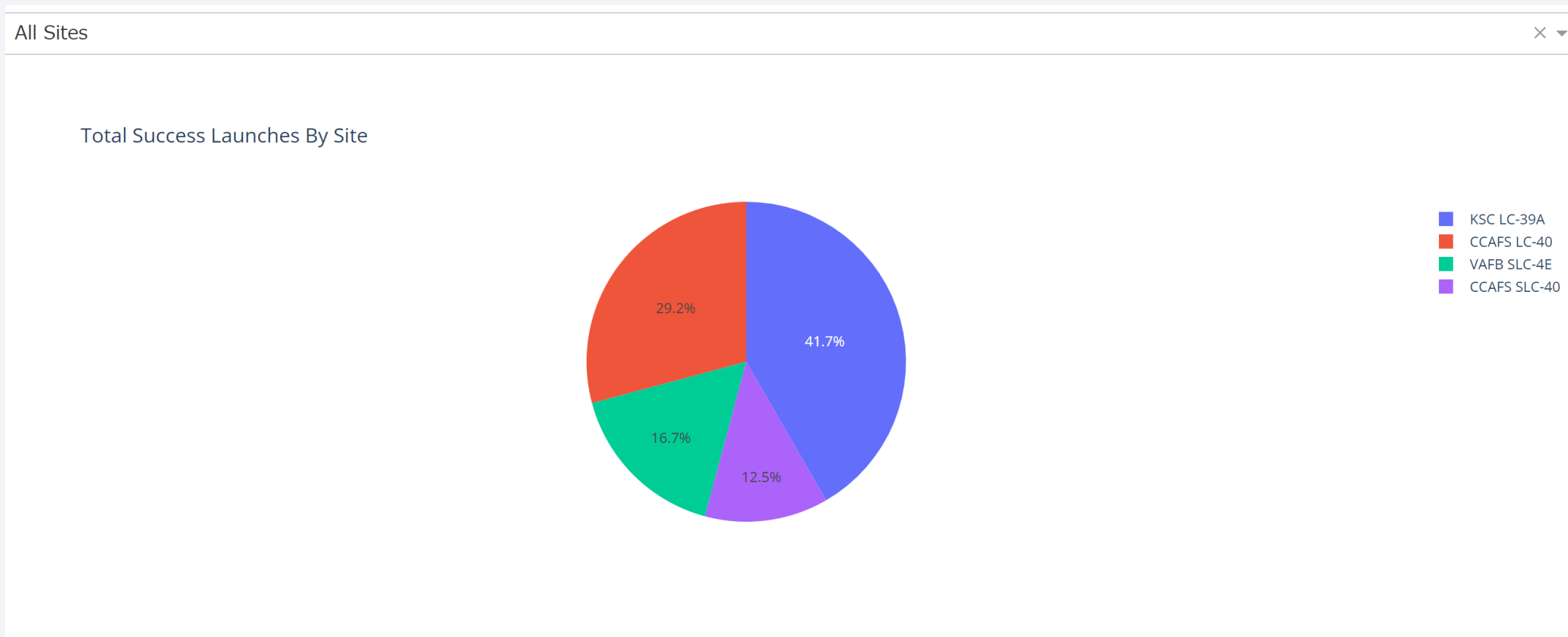


Section 4

# Build a Dashboard with Plotly Dash

# <Dashboard Screenshot 1>

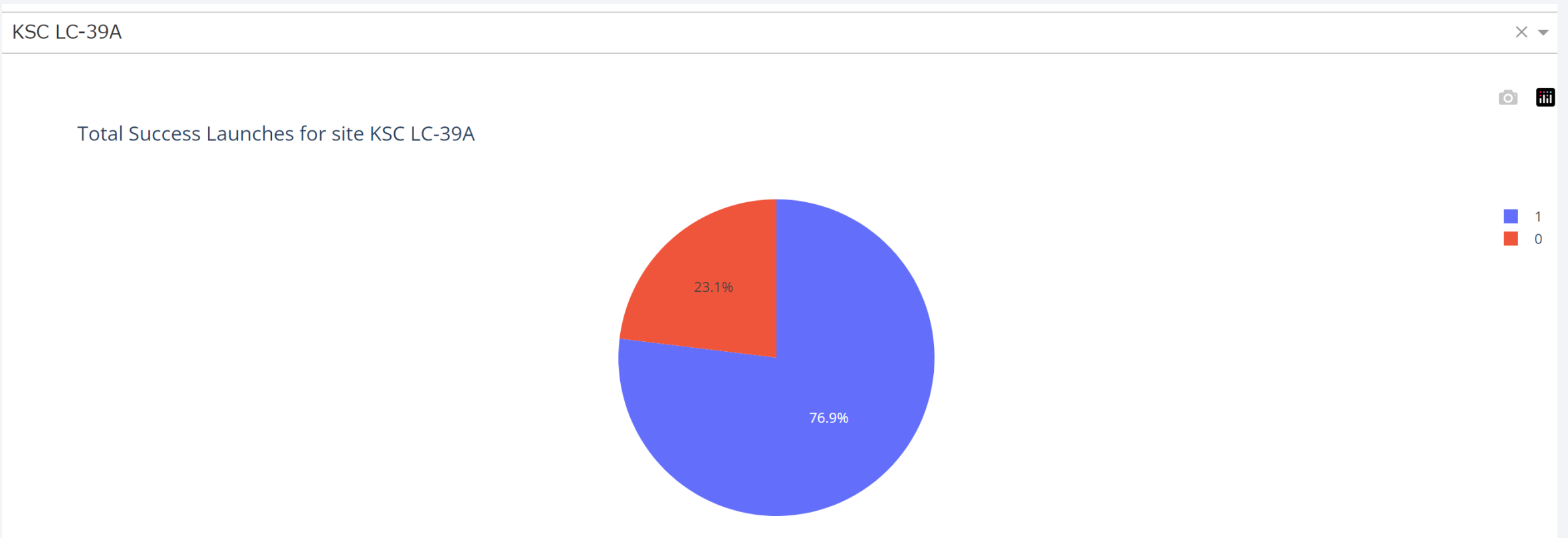
---



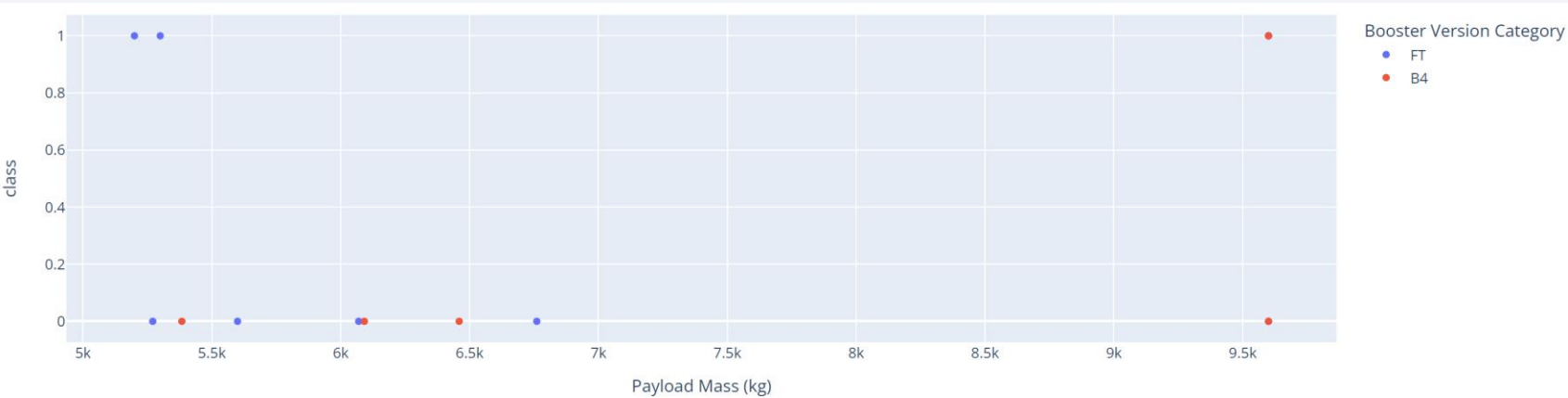
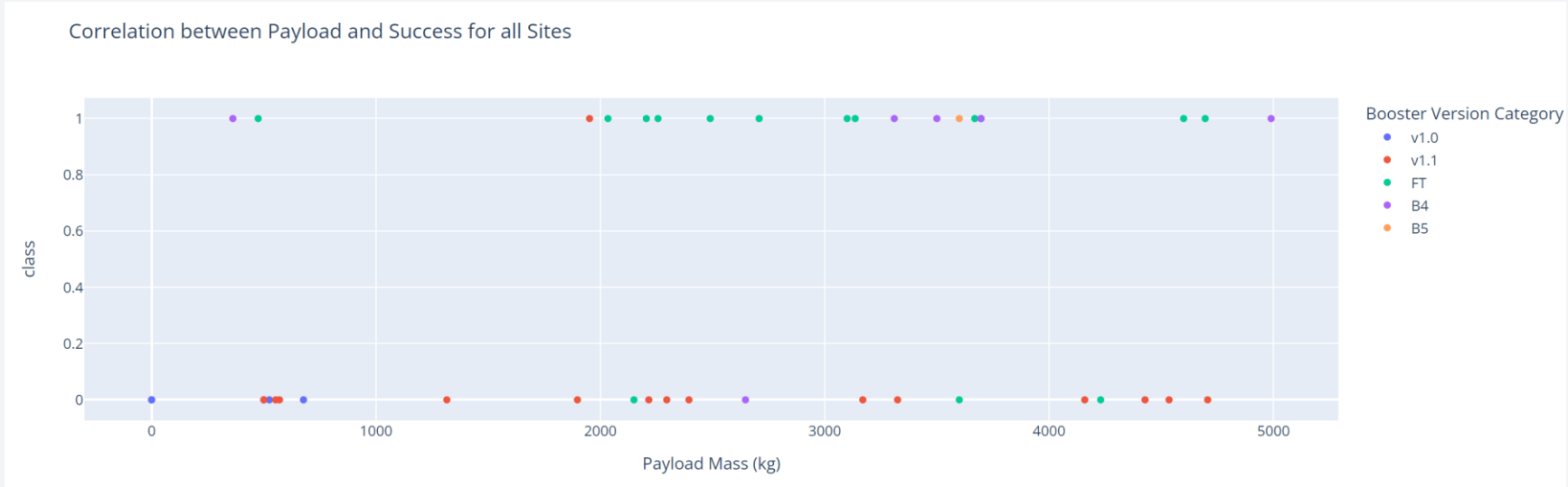


# <Dashboard Screenshot 2>

---



# <Dashboard Screenshot 3>



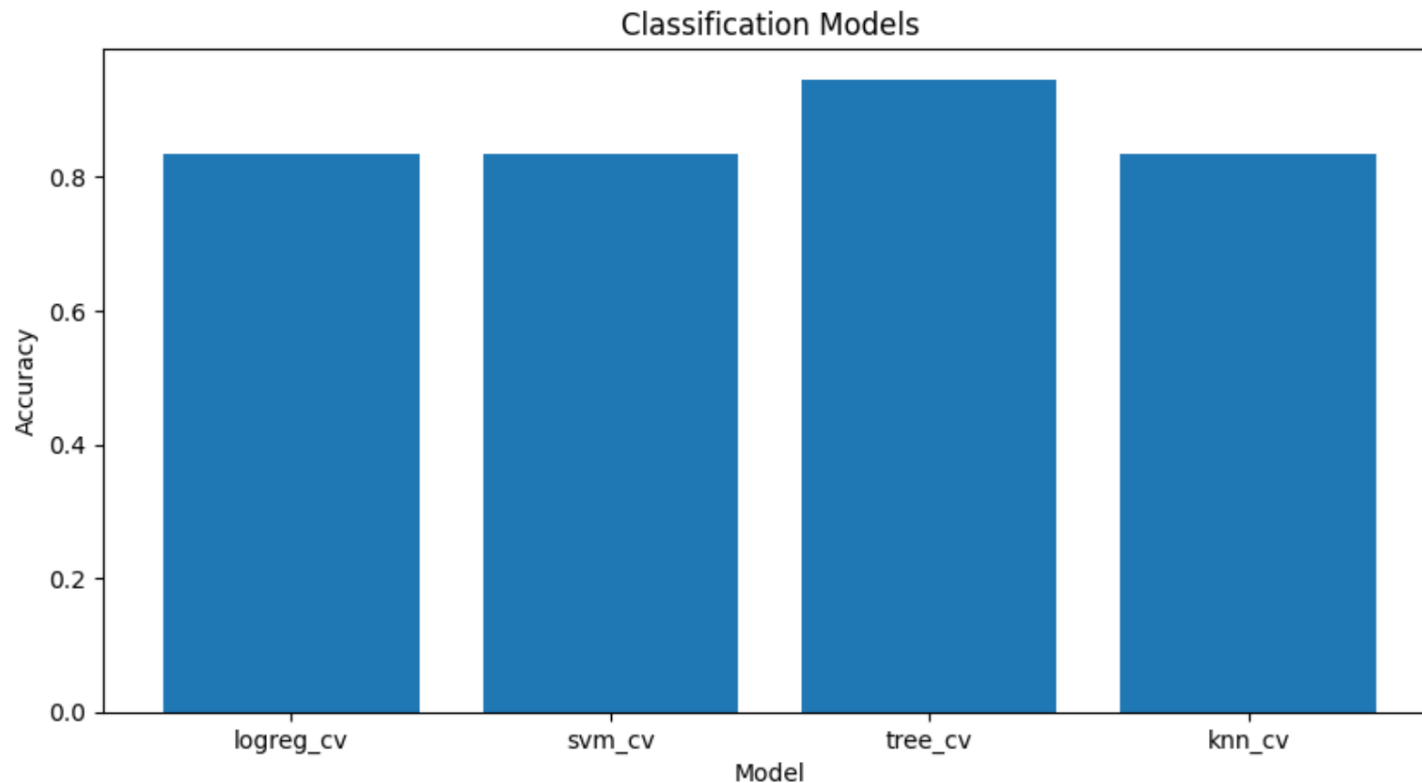
Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)  
print("accuracy :",tree_cv.best_score_)
```

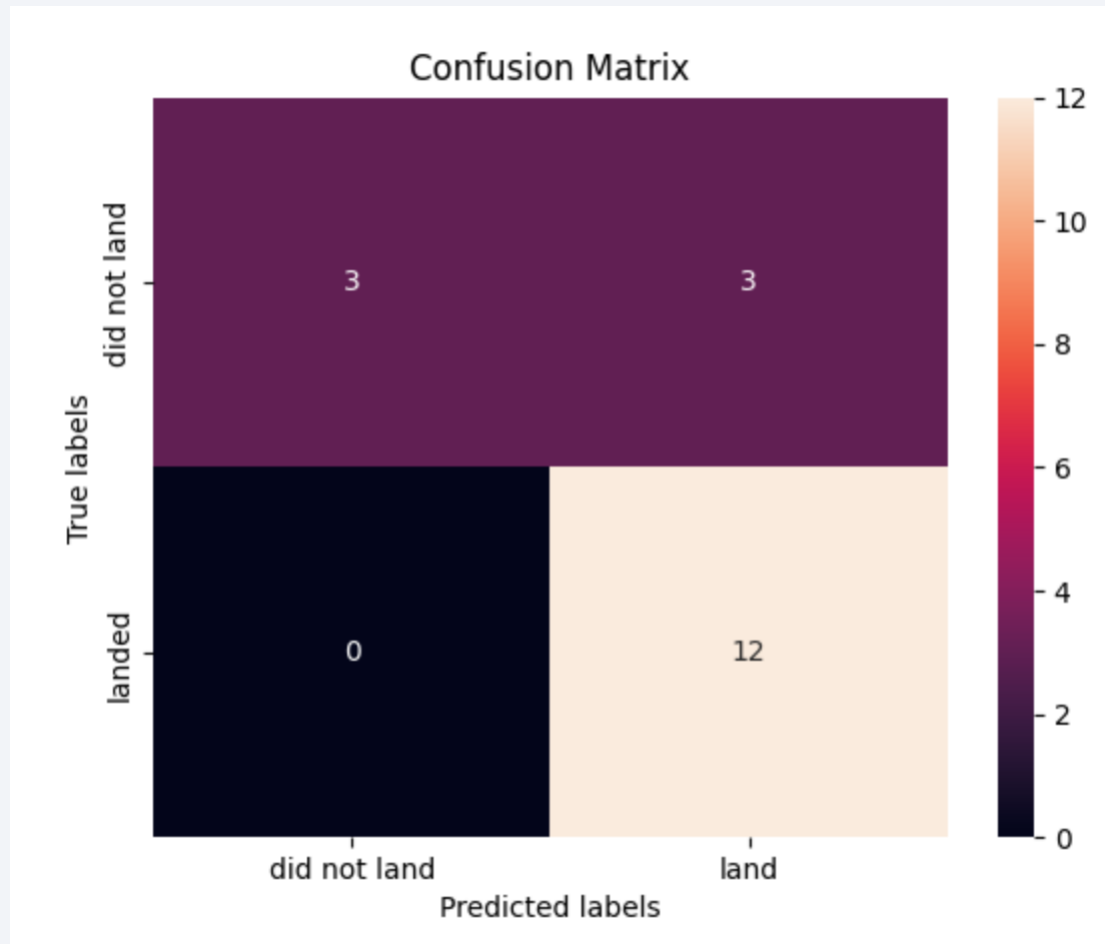
```
tuned hpyerparameters :(best parameters) {'criterion': 'gini', 'max_depth': 18,  
'min_samples_split': 10, 'splitter': 'best'}  
accuracy : 0.875
```



- 最も精度が高いのは`tree_cv`で、**0.875**の精度です。
- `logreg_cv`、`svm_cv`、`knn_cv`の精度は**0.8**程度です。

# Confusion Matrix

Confusion Matrixの結果は、すべてのケースで次のようになりました。



- 偽陽性の値は0
- 真陽性と偽陰性の値は3
- 真性陰性の値は12

# Conclusions

---

- 便数が増えるほど成功率が上がる傾向がある
- **2019**年以降は成功率が**80%**以上となる
- ペイロード質量が**8000kg**以上の時に打ち上げるのが良い
- ペイロードが**6000kg**未満の場合、最も成功率の高い軌道は**SSO**である
- 打ち上げ場所が**KSC LC-9A**の場合、成功率が高くなる
- ペイロード質量が**5000kg**未満の場合、ブースターバージョンの分類は**FT**とする
- 分類精度が最も高いのは**tree\_cv**
- 混合行列では偽陰性は**0**なので信頼度は高い



Thank you!

